Goldstone DSCC Energy Distribution Model

B. H. Chapman
TDA Planning Office

In expectation of increases in cost and decreases in supply of currently available energy forms, the DSN is studying the installation of systems which will provide reliable Deep Space Communications Complex energy in stable amounts and at stable cost. One of the main factors in improving the economic viability of such an installation is the efficiency with which the useful energy forms resulting from the conversion of the stable energy form to be provided can be distributed to the consumers. The aim of the following general distribution model is to provide a method for the optimal design of a network for the distribution of several different types of energy to users and for the optimal operation of such a network when installed. When such a network is operational the consumers' demand for energy can be ascertained by real-time sampling but during the design phase these energy demands are known only stochastically. The initial model below describes the case of known constant demand and will form the basis of a subsequent model of the stochastic demand case. An algorithm to be used in the solution of this model problem is also outlined.

I. Mathematical Model I

In detail the situation to be modeled is as follows: Several plants and the related distribution system are to be constructed to serve the electrical, heating, and cooling needs of several established energy consumers whose demand for each energy form is constant and known. The possible locations of the plants are given, but the particular sites to be used are to be selected so as to result in the least total construction and operational cost. Each plant has a known cost function of its electrical capacity, reflecting the initial capital cost plus the operational and maintenance cost over the expected lifetime of the plant. The heat output of each plant is the sum of two terms: (1) recovered waste heat, which is a known function of the plant's electrical output, and (2) heat obtained from fuel directly at a known cost. A portion of this heat output is then converted at a known cost and efficiency to chilled water for use in cooling. Each link of the distribution system has a known cost function of its capacity, and the losses in each link are known functions of the amount of energy traversing that link.

If a particular link is to be constructed, its capacity must lie between prescribed upper and lower bounds. The distribution system also allows for the transfer of electrical energy between plants, intermediate energy distribution nodes at various sites between plants and consumers, and substitution of electrical energy for heating and cooling energy at some known efficiencies and costs. Thus the problem is to determine which sites are to be used, what capacity plant to install at each selected site, and what the distribution pattern should be for each energy form. This is to be done so as to minimize the total construction and operational cost while satisfying the demands of each consumer and the capacity constraints on the distribution system.

In order to maintain a clear relationship between the physical problem and the following mathematical formulation, the variables and functions involved will all be triply subscripted. The first subscript refers to the level of the distribution system with which the variable or function is associated. The second and third subscripts refer, respectively, to the origin and destination within that level of the quantity described by the variable or function (Fig. 1).

A. System Variables

1. Level 0: Plant Variables. There are k possible sites for total energy plants (TEP) which derive electricity, heat, and chilled water from fuel. The variables are:

e_{00i} = electrical output of plant i

$H_{00i}(e_{00i})$ = recovered heat output function of plant i

h_{00i} = directly derived heat output of plant i

h_{0ii} = portion of heat output of plant i used for conversion to chilled water

$C_{0ii}(h_{0ii})$ = chilled water output function of plant i

e_{0ij} = electrical energy leaving plant i for plant j

$E_{0ij}(e)$ = electrical energy arriving at plant j when electrical energy e originated from plant i

2. Level 1: Primary Distribution Variables. The amounts of the three energy types leaving the plants and arriving at the ℓ electrical consumers, m heating energy consumers, and n cooling energy consumers are:

e_{1ij} = electrical energy leaving plant i for electrical consumer j

$E_{1ij}(e)$ = electrical energy arriving at electrical consumer j when electrical energy e originated from plant i

c_{1ij} = cooling energy leaving plant i for cooling energy consumer j

$C_{1ij}(c)$ = cooling energy arriving at cooling energy consumer j when cooling energy c originated from plant i

h_{1ij} = heating energy leaving plant i for heating energy consumer j

$H_{1ij}(h)$ = heating energy arriving at heating energy consumer j when heating energy h originated from plant i

3. Level 2: Secondary Distribution Variables. Subsequent amounts of the three energy types distributed between consumers of the same energy type. The variables are:

e_{2ij} = electrical energy leaving electrical consumer i for electrical consumer j

$E_{2ij}(e)$ = electrical energy arriving at electrical consumer j when electrical energy e originated from electrical consumer i

c_{2ij} = cooling energy leaving cooling energy consumer i for cooling energy consumer j

$C_{2ij}(c)$ = cooling energy arriving at cooling energy consumer j when cooling energy c originated from cooling energy consumer i

h_{2ij} = heating energy leaving heating energy consumer i for heating energy consumer j

$H_{2ij}(h)$ = heating energy arriving at heating energy consumer j when heating energy h originated from heating energy consumer i

4. Level 3: Cooling Energy Substitution. Amounts of electrical energy from each electrical consumer used for substitution as cooling energy at each cooling energy consumer are:

e_{3ij} = electrical energy leaving electrical consumer i for substitution as cooling energy at cooling energy consumer j

$C_{3ij}(e)$ = cooling energy arriving at cooling energy consumer j when electrical energy e originated from electrical consumer i for substitution

5. **Level 4: Heating Energy Substitution.** Amounts of electrical energy from each electrical consumer used for substitution as heating energy at each heating energy consumer are:

e_{4ij} = electrical energy leaving electrical consumer i for substitution as heating energy at heating energy consumer j

$H_{4ij}(e)$ = heating energy arriving at heating energy consumer j when electrical energy e originated from electrical consumer i for substitution

B. Demands

e_j = electrical demand of electrical consumer j

c_j = cooling demand of cooling energy consumer j

h_j = heating demand of heating energy consumer j

If the energy input to a heating or cooling energy consumer exceeds the demand, the excess energy is dumped as waste. In order to keep the total system energy constant, this wasted energy is accounted for by the following *slack variables*:

c'_j = waste cooling energy dumped at cooling energy consumer j

h'_j = waste heating energy dumped at heating energy consumer j

(If consumer j is actually a dummy consumer representing a possible distribution node for electrical, cooling, or heating energy, then e_j or c_j and c'_j or h_j and h'_j, respectively, are set equal to 0).

C. Cost Functions

$\alpha_{00i}(e)$ = cost of installing electrical generation capacity e and waste heat recovery of capacity $H(e)$ at plant i

$\alpha_{0ij}(e)$ = cost of installing an electrical link of capacity e between plant i and plant j

$\alpha_{1ij}(e)$ = cost of installing an electrical link of capacity e between plant i and electrical consumer j

$\alpha_{2ij}(e)$ = cost of installing an electrical link of capacity e between electrical consumer i and electrical consumer j

$\alpha_{3ij}(e)$ = cost of installing an electrical substitution link taking electrical energy e from electrical consumer i to cooling energy consumer j and converting it to provide cooling energy $C_{3ij}(e)$

$\alpha_{4ij}(e)$ = cost of installing an electrical substitution link taking electrical energy e from electrical consumer i to heating energy consumer j and converting it to provide heating energy $H_{4ij}(e)$

$\beta_{00i}(h)$ = cost of installing direct heat generation of capacity h at plant i

$\beta_{0ii}(h)$ = cost of installing heat to chilled water conversion of capacity h at plant i to provide chilled water output $C_{0ii}(h)$

$\beta_{1ij}(h)$ = cost of installing a heating link of capacity h between plant i and heating energy consumer j

$\beta_{2ij}(h)$ = cost of installing a heating link of capacity h between heating energy consumer i and heating energy consumer j

$\gamma_{1ij}(c)$ = cost of installing a cooling link of capacity c between plant i and cooling energy consumer j

$\gamma_{2ij}(c)$ = cost of installing a cooling link of capacity c between cooling energy consumer i and cooling energy consumer j

All of the above functions include initial capital cost and installation cost plus the expected maintenance cost over the expected system lifetime.

II. Linearization of Functions

In general, all of the above cost, loss, and production functions will be nonlinear. In order to make the mathematical problem more tractable by current computational techniques, it is desirable to replace each of these functions by an approximating piecewise-linear function. This can be done to any desirable accuracy since, in general, the functions being approximated will be at least piecewise smooth. For example, consider the recovered heat and cost functions of the electrical generation capacity of a plant at site i (Fig. 2). It is then possible to approximate $H_{00i}(e_{00i})$ and $\alpha_{00i}(e_{00i})$ to within allowable error by functions which are linear on the intervals e^0_{00ik}, e^1_{00ik} for $k = 1 \ldots \ell_{00i}$ as in Fig. 3.

The partition values e^0_{00ik} and e^1_{00ik} and the approximating linear functions can be determined for example by a piecewise-linear least squares fit simultaneously on both functions.

The single plant in the description above with its upper and lower bounds on electrical generation capacity is then replaced by $\ell_{00i} = 3$ "pseudo-plants" with the relevant upper and lower bounds to describe the capacity region in which each should operate.

The dichotomous variable λ_{00ik} is then introduced such that:

$\lambda_{00ik} = 1$ if pseudo-plant k is actually to be constructed at plant site i

$\lambda_{00ik} = 0$ if pseudo-plant k is not to be constructed

Since at most one of these pseudo-plants will be constructed at plant site i, we have the additional constraint:

$$\sum_{k=1}^{\ell_{00i}} \lambda_{00ik} \leqslant 1$$

Since the capacity of pseudo-plant k should be zero if it is not to be constructed and it should not be constructed if its capacity is zero, we have the following constraint to force the correct logical relationship between the dichotomous and capacity variables:

$$\lambda_{00ik}\, e^{0}_{00ik} \leqslant e_{00ik} \leqslant \lambda_{00ik}\, e^{1}_{00ik}$$

Each pseudo-plant will then have linear heat-recovery and cost functions in its limited capacity range:

$$H_{00ik}(e_{00ik}) = H_{00ik}\, e_{00ik} + H^{*}_{00ik} = H_{00ik}\, e_{00ik} + H^{*}_{00ik}\, \lambda_{00ik}$$

$$\alpha_{00ik}(e_{00ik}) = \alpha_{00ik}\, e_{00ik} + \alpha^{*}_{00ik} = \alpha_{00ik}\, e_{00ik} + \alpha^{*}_{00ik}\, \lambda_{00ik}$$

The same artifice may be used to piecewise-linearize each nonlinear function appearing in the model. This introduces the following bounds on the allowed operating capacities of the pseudo-system elements to be defined in doing so.

III. Upper and Lower Bounds on Capacities of Pseudo-system Elements

$e^{1}_{00ik}, e^{0}_{00ik}$ = upper and lower bounds on electrical capacity of pseudo-plant k at plant site i, if built

$h^{1}_{00ik}, h^{0}_{00ik}$ = upper and lower bounds on direct heat generation capacity of pseudo-plant k at plant site i, if utilized

$h^{1}_{0iik}, h^{0}_{0iik}$ = upper and lower bounds on capacity of heat to chilled water conversion of pseudo-plant k at plant site i, if utilized

$e^{1}_{0ijk}, e^{0}_{0ijk}$ = upper and lower bounds on capacity of pseudo-electrical link k between plant site i and plant site j if built

$e^{1}_{1ijk}, e^{0}_{1ijk}$ = upper and lower bounds on capacity of pseudo-electrical link k between plant site i and electrical consumer j if built

$c^{1}_{1ijk}, c^{0}_{1ijk}$ = upper and lower bounds on capacity of pseudo-cooling link k between plant site i and cooling energy consumer j if built

$h^{1}_{1ijk}, h^{0}_{1ijk}$ = upper and lower bounds on capacity of pseudo-heating link k between plant site i and heating energy consumer j if built

$e^{1}_{2ijk}, e^{0}_{2ijk}$ = upper and lower bounds on capacity of pseudo-electrical link k between electrical consumer i and electrical consumer j if built

$c^{1}_{2ijk}, c^{0}_{2ijk}$ = upper and lower bounds on capacity of pseudo-cooling link k between cooling energy consumer i and cooling energy consumer j if built

$h^{1}_{2ijk}, h^{0}_{2ijk}$ = upper and lower bounds on capacity of pseudo-heating link k between heating energy consumer i and heating energy consumer j if built

$e^{1}_{3ijk}, e^{0}_{3ijk}$ = upper and lower bounds on capacity of pseudo-electrical substitution link k from electrical consumer i to cooling energy consumer j if utilized

$e^{1}_{4ijk}, e^{0}_{4ijk}$ = upper and lower bounds on capacity of pseudo-electrical substitution link k from electrical consumer i to heating energy consumer j if utilized

As in the above example, the construction or nonconstruction of each pseudo-system element is controlled by the use of dichotomous variables.

$\lambda_{00ik} = 1$ if pseudo-plant k at plant site i is to be used for electrical generation $(k = 1 \ldots \ell_{00i})$

$\mu_{00ik} = 1$ if pseudo-plant k at plant site i is to be used for direct heat generation $(k = 1 \ldots m_{00i})$

$\mu_{0iik} = 1$ if pseudo-plant k at plant site i is to be used for production of chilled water $(k = 1 \ldots m_{0ii})$

$\lambda_{0ijk} = 1$ if plant site i is to be connected to plant site j by a pseudo-electrical link with capacity between e^{0}_{0ijk} and e^{1}_{0ijk} $(k = 1 \ldots \ell_{0ij})$

$\lambda_{1ijk} = 1$ if plant site i is to be connected to electrical consumer j by a pseudo-electrical link with capacity between e^0_{1ijk} and e^1_{1ijk} $(k = 1 \ldots \ell_{1ij})$

$\mu_{1ijk} = 1$ if plant site i is to be connected to heating energy consumer j by a pseudo-heating link with capacity between h^0_{1ijk} and h^1_{1ijk} $(k = 1 \ldots m_{1ij})$

$\nu_{1ijk} = 1$ if plant site i is to be connected to cooling energy consumer j by a pseudo-cooling link with capacity between c^0_{1ijk} and c^1_{1ijk} $(k = 1 \ldots n_{1ik})$

$\lambda_{2ijk} = 1$ if electrical consumer i is to be connected to electrical consumer j by a pseudo-electrical link with capacity between e^0_{2ijk} and e^1_{2ijk} $(k = 1 \ldots \ell_{2ij})$

$\mu_{2ijk} = 1$ if heating energy consumer i is to be connected to heating energy consumer j by a pseudo-heating link with capacity between h^0_{2ijk} and h^1_{2ijk} $(k = 1 \ldots m_{2ij})$

$\nu_{2ijk} = 1$ if cooling energy consumer i is to be connected to cooling energy consumer j by a pseudo-cooling link with capacity between c^0_{2ijk} and c^1_{2ijk} $(k = 1 \ldots n_{2ij})$

$\lambda_{3ijk} = 1$ if substitution from electrical consumer i to cooling energy consumer j is to be provided by a pseudo-electrical substitution link with capacity between e^0_{3ijk} and e^1_{3ijk} $(k = 1 \ldots \ell_{3ij})$

$\lambda_{4ijk} = 1$ if substitution from electrical consumer i to heating energy consumer j is to be provided by a pseudo-electrical substitution link with capacity between e^0_{4ijk} and e^1_{4ijk} $(k = 1 \ldots \ell_{4ij})$

The constant demand problem can then be formulated as the following mixed integer linear program:

Minimize

$$\sum_{i=0}^{4}\sum_{j}\sum_{k}\sum_{\ell=1}^{\ell_{ijk}} \{\alpha_{ijk\ell}e_{ijk\ell} + \alpha^*_{ijk\ell}\lambda_{ijk\ell}\}$$
$$+ \sum_{i=0}^{2}\sum_{j}\sum_{k}\sum_{\ell=1}^{m_{ijk}} \{\beta_{ijk\ell}h_{ijk\ell} + \beta^*_{ijk\ell}\mu_{ijk\ell}\}$$
$$+ \sum_{i=1}^{2}\sum_{j}\sum_{k}\sum_{\ell=1}^{n_{ijk}} \{\gamma_{ijk\ell}c_{ijk\ell} + \gamma^*_{ijk\ell}\nu_{ijk\ell}\}$$

where the sums over j and k are taken over all possible combinations corresponding to the value of the distribution level subscript i and the type of the variable (e, h, or c).

This will give the minimum *total* cost of the system with the following constraints:

Subject to

$$\lambda_{ijk\ell}, \mu_{ijk\ell}, \nu_{ijk\ell} = 0 \text{ or } 1$$

$$h'_j, c'_j, e_{ijk\ell}, h_{ijk\ell}, c_{ijk\ell} \geq 0$$

The following constraints arise from the energy balance (energy flowing in equals energy flowing out) at each node of the distribution system:

$$\sum_{K=1}^{\ell_{00j}} e_{00jK} + \sum_{\substack{i=1\\i\neq j}}^{k}\sum_{K=1}^{\ell_{0ij}} \{E_{0ijK}e_{0ijK} + E^*_{0ijK}\lambda_{0ijK}\}$$
$$= \sum_{\substack{r=1\\r\neq j}}^{k}\sum_{K=1}^{\ell_{0jr}} e_{0jrK} + \sum_{r=1}^{\ell}\sum_{K=1}^{\ell_{1jr}} e_{1jrK} \qquad \text{for } j = 1 \ldots k$$

$$\sum_{K=1}^{\ell_{00j}} \{H_{00jK}e_{00jK} + H^*_{00jK}\lambda_{00jK}\} + \sum_{K=1}^{m_{00j}} h_{00jK}$$
$$= \sum_{K=1}^{m_{0jj}} h_{0jjK} + \sum_{r=1}^{m}\sum_{=1}^{m_{1jr}} h_{1jrK} \qquad \text{for } j = 1 \ldots k$$

$$\sum_{K=1}^{m_{0jj}} \{C_{0jjK}h_{0jjK} + C^*_{0jjK}\mu_{0jjK}\}$$
$$= \sum_{r=1}^{n}\sum_{K=1}^{n_{1jr}} c_{1jrK} \qquad \text{for } j = 1 \ldots k$$

$$\sum_{i=1}^{k}\sum_{K=1}^{\ell_{1ij}} \{E_{1ijK}e_{1ijK} + E^*_{1ijK}\lambda_{1ijK}\}$$
$$+ \sum_{\substack{i=j\\i\neq j}}^{\ell}\sum_{K=1}^{\ell_{2ij}} \{E_{2ijK}e_{2ijK} + E^*_{2ijK}\lambda_{2ijK}\}$$
$$= e_j + \sum_{\substack{r=1\\r\neq j}}^{\ell}\sum_{K=1}^{\ell_{2jr}} e_{2jrK} + \sum_{r=1}^{n}\sum_{K=1}^{\ell_{3jr}} e_{3jrK} + \sum_{r=1}^{m}\sum_{K=1}^{\ell_{4jr}} e_{4jrK}$$
$$\text{for } j = 1 \ldots \ell$$

$$\sum_{i=1}^{k}\sum_{K=1}^{m_{1ij}} \{H_{1ijK}h_{1ijK} + H^*_{1ijK}\mu_{1ijK}\}$$
$$+ \sum_{\substack{i=1\\i\neq j}}^{m}\sum_{K=1}^{m_{2ij}} \{H_{2ijK}h_{2ijK} + H^*_{2ijK}\mu_{2ijK}\}$$
$$+ \sum_{i=1}^{\ell}\sum_{K=1}^{\ell_{4ij}} \{H_{4ijK}e_{4ijK} + H^*_{4ijK}\lambda_{4ijK}\}$$
$$= h_j + h'_j + \sum_{\substack{r=1\\r\neq j}}^{m}\sum_{K=1}^{m_{2jr}} h_{2jrK} \qquad \text{for } j = 1 \ldots m$$

$$
\begin{aligned}
&\sum_{i=1}^{k} \sum_{\kappa=1}^{n_{1ij}} \{C_{1ij\kappa}c_{1ij\kappa} + C^{*}_{1ij\kappa}\nu_{1ij\kappa}\} \\
&\quad + \sum_{\substack{i=1 \\ i \neq j}}^{n} \sum_{\kappa=1}^{n_{2ij}} \{C_{2ij\kappa}c_{2ij\kappa} + C^{*}_{2ij\kappa}\nu_{2ij\kappa}\} \\
&\quad + \sum_{i=1}^{\ell} \sum_{\kappa=1}^{\ell_{1ij}} \{C_{3ij\kappa}e_{3ij\kappa} + C^{*}_{3ij\kappa}\lambda_{3ij\kappa}\} \\
&\quad = cj + c'_j + \sum_{\substack{r=1 \\ r \neq j}}^{n} \sum_{\kappa=1}^{n_{2jr}} c_{2jr\kappa} \qquad \text{for } j = 1 \ldots n
\end{aligned}
$$

The following constraints, besides keeping the capacity of each pseudo-system element between the relevant upper and lower bounds if it is to be constructed, also force the correct logical relationships between the capacity variables and the associated dichotomous variables.

$$
\begin{aligned}
\lambda_{ij\kappa\ell}\, e^{0}_{ij\kappa\ell} &\leqslant e_{ij\kappa\ell} \leqslant \lambda_{ij\kappa\ell}\, e^{1}_{ij\kappa\ell} \\
\mu_{ij\kappa\ell}\, h^{0}_{ij\kappa\ell} &\leqslant h_{ij\kappa\ell} \leqslant \mu_{ij\kappa\ell}\, h_{ij\kappa\ell} \\
\nu_{ij\kappa\ell}\, c^{0}_{ij\kappa\ell} &\leqslant c_{ij\kappa\ell} \leqslant \nu_{ij\kappa\ell}\, c^{1}_{ij\kappa\ell}
\end{aligned}
$$

where the subscripts i, j, κ, and ℓ run over all allowable combinations, depending on the type of variable.

The remaining set of constraints insures that at most one of the possible pseudo-system elements in each case is to be constructed as an element of the real system.

$$
\sum_{\ell=1}^{\ell_{ij\kappa}} \lambda_{ij\kappa\ell} \leqslant 1
$$

$$
\sum_{\ell=1}^{m_{ij\kappa}} \mu_{ij\kappa\ell} \leqslant 1
$$

$$
\sum_{\ell=1}^{n_{ij\kappa}} \nu_{ij\kappa\ell} \leqslant 1
$$

where the subscripts i, j, and κ run over all allowable combinations depending on the type of the associated capacity variable.

IV. Method of Solution

The most favorable results in solving large mixed integer linear programs like the above are currently given by branch and bound methods (or specialized methods which utilize branch and bound methods as part of their procedure.) The method is a search procedure which estimates or evaluates the maximum objective function value for all possible combinations of values of the integer restricted variables. It begins with a large set of possible combinations of values for the integer-restricted variables and then divides this set into successively smaller subsets.

At each step an estimate is made of the maximum objective function value given that the combination of values of the integer restricted variables lies in each subset. Also at each step this estimate of the maximum objective function value for each subset is compared with the objective function value of a solution which satisfies the constraints and has integer values for the integer-restricted variables. Subsets whose maximum objective function values cannot exceed the value of the best current integer solution are then no longer considered as candidates for containing the optimum combination of values of the integer-restricted variables.

If at any step an integer solution is found whose evaluated objective function value is larger than that of the current best integer solution then it is taken as the updated best integer solution. Continuing in this manner the subsets are partitioned more and more finely and are eliminated as their maximum possible objective function values fall below the increasing objective function value of the best current integer solution. Eventually a point is reached where one of the subsets will contain only the optimum combination of values of the integer-restricted variables and this solution will then become the current best integer solution. From this point on, the comparison of objective function values will eliminate all the remaining subsets of possible combinations of values of the integer-restricted variables and establish this solution as the true optimum.

In more detail, the method is illustrated by the accompanying structured Level 1 flowchart. When the given mixed-integer linear program is feasible, it is solved as a linear program neglecting the integer constraints to obtain the objective function value $\bar{a}_{00}$. Prior to solving this problem there was no current feasible solution in which the integer-restricted variables took on integer values; so the objective function value of the current best integer solution x_{0c} is set equal to $-\infty$. Likewise before this problem was solved there was no estimate of an upper bound on its objective function value so UB_P was set equal to ∞.

If the solution of this problem is such that all integer-restricted variables have integer values, the optimal solution has been found immediately. If not, one of the variables whose integer constraint is not satisfied in the current solution is chosen for the branching process. Here two new subproblems are created by restricting the value of the unsatisfied variable to be greater than the integer immediately above its current value and to be less than the integer immediately below its current value. The initial feasible region is then divided into two disjoint regions, one of which must contain the optimum solution, since the branching variable must be integer-valued and only a non-integer portion of feasible region has been removed between these two disjoint regions. One of these subproblems is then chosen to be solved immediately as a linear programming problem neglecting the integer constraints. The other subproblem is stored in a list with an upper bound on its objective function value to be solved later. This process is then repeated with the solved subproblem becoming the current problem at each step until one of its branched descendants yields a solution whose integer-restricted variables have integer values. The value of the objective function of this solution is recorded as x_{0c} and any subproblem in the stored list whose upper bound is less than x_{0c} can be eliminated since its feasible region could not have contained the optimum solution. The method then backtracks by choosing a problem from the stored list to begin the procedure again.

At each branching step the feasible region is split into two disjoint regions and the non-integer region between them is removed from the feasible region. Hence, at any point in the procedure, exactly one of the subproblems contains the optimal solution (if it is unique). In cases where one of the integer restricted variables is not constrained above but the problem does have a finite optimum solution, the method will keep reducing the infinite portion of the feasible region until the upper bound on the objective function value associated with that region falls below the objective function value of the current best integer solution at which time it can be eliminated. This leaves only the disjoint finite feasible regions which the procedure continues to divide while eliminating the non-integer regions of the integer restricted variables until the objective function values of all remaining feasible regions have been examined and the optimum solution found or infeasibility demonstrated.

The details of the methods used in the branching and backtracking subroutines to determine which of the problems is to be solved next and in finding upper bounds on the objective function values are given in an appendix. This procedure has been used successfully on large mixed integer programming problems with on the order of one hundred 0-1 integer variables and several thousand continuous variables (Refs. 1, 2), and hence will be effective in dealing with problems having a few consumers and plants and involving mildly non-linear cost and production functions.

In cases where there are many consumers and plants and more non-linear functions, however, the number of 0-1 integer variables increases enormously to the point where branch and bound methods cannot solve the problem within a reasonable amount of computer time even if many of the obviously uneconomical combinations of values of these variables have been eliminated beforehand. In cases such as this, a refined procedure utilizing "Bender's decomposition" can be used. This is an iterative procedure which at each step deals only with decoupled problems describing the distribution of individual commodities (here, energy types). Besides being able to deal with much larger problems (problems having about two thousand 0-1 integer variables and about twenty thousand continuous variables have been solved (Ref. 3)). The fact that at each step the procedure deals with only the usual "transportation problem" involving a single commodity allows the problem of stochastic demand for these commodities to be dealt with much more simply than the case where the transportation problems are coupled together.

An important problem for DSN energy distribution is then the formulation of the full stochastic problem in terms of a Bender's decomposition algorithm.

References

1. Forrest, J. J. H., Hirst, J. P. H., and Tomlin, J. A., "Practical Solution of Large Mixed Integer Programming Problems," in *Management Science*, Vol. 20, No. 5, p. 736, 1974.

2. Tomlin, J. A., "Branch and Bound Methods for Integer and Non-convex Programming," in *Integer and Non-linear Programming*, p. 437, edited by J. Abadie. North Holland Pub. Co., Amsterdam, 1970.

3. Geoffrion, A. M., and Graves, G. W., "Multicommodity Distribution System Design," in *Management Science*, Vol. 20, No. 5, p. 822, 1974.

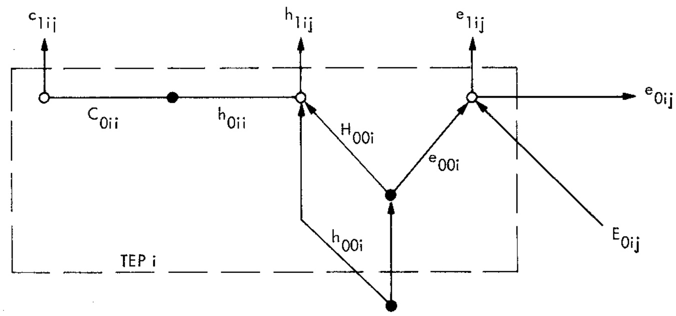
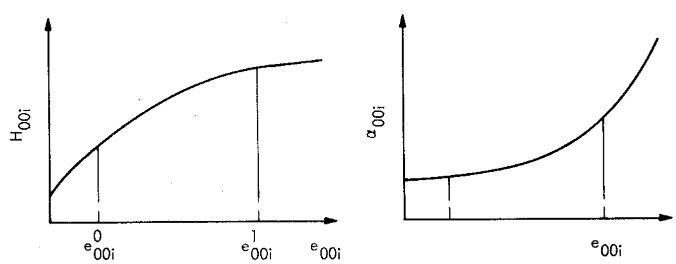


Fig. 1. Total energy plant



e^0_{00i} AND e^1_{00i} = ORIGINAL LOWER AND UPPER BOUNDS OF e_{00i}

Fig. 2. Original recovered heat and cost functions

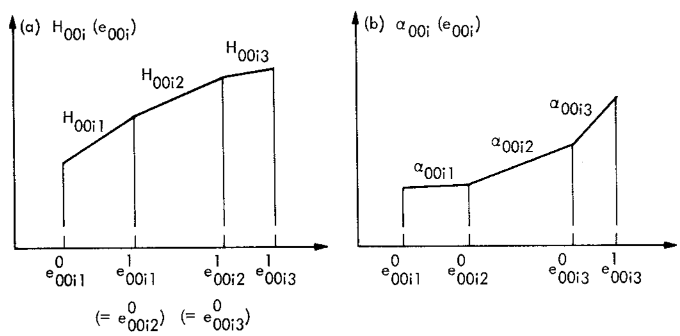


Fig. 3. Piecewise linear approximations to recovered heat and cost functions

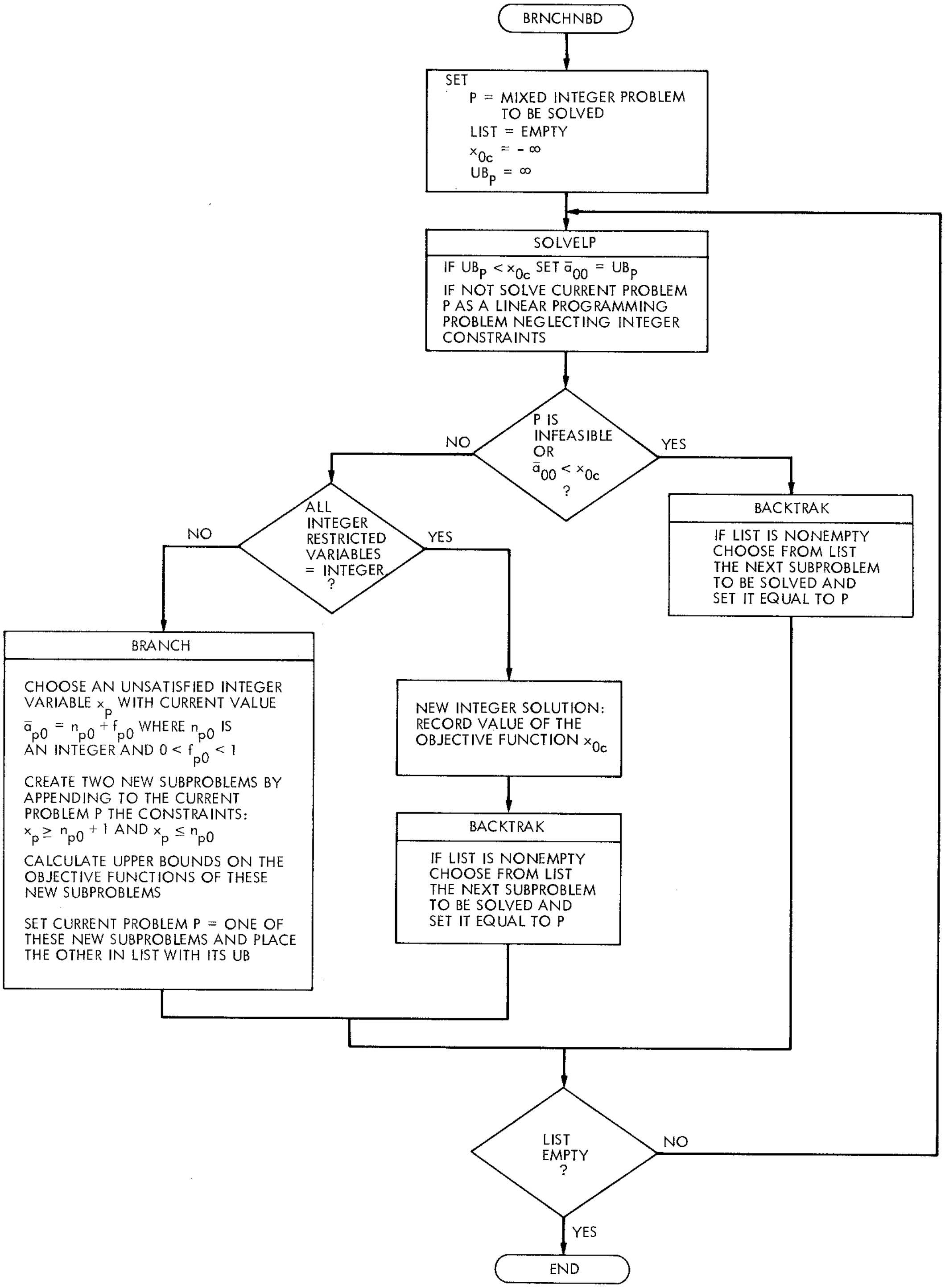


Fig. 4. Flowchart for BRNCHNBD

Appendix A

Details of Methods Which Can Be Used in BRANCH and BACKTRAK Routines

I. BRANCH

In the optimal solution of the current linear programming (LP) subproblem some of the integer-restricted-variables will take on non-integer values. In order to force these variables toward integer values, two more tightly constrained subproblems are formed. Let the integer-restricted variable x_p have $\bar{a}_{p0} = n_{p0} + f_{p0}$ (where n_{p0} is an integer and $0 \leqslant f_{p0} < 1$) as its value in the current optimal solution. The feasible set for the current subproblem is then reduced by appending the further constraint: $x_p \geqslant n_{p0} + 1$ to yield one subproblem and by appending $x_p \leqslant n_{p0}$ to yield the other subproblem. The decision of which unsatisfied integer variable to choose for this branching process is often based on the calculation of *penalties* which estimate the change in objective function value due to the newly appended constraints.

A. Simplex Algorithm

As a preface to the following procedures, some of the salient points of the simplex algorithm will be described. Suppose we have the standard linear programming problem:

Maximize the linear form

$$\sum_{j=1}^{n} \bar{a}_{0j} x_j$$

where the $\bar{a}_{0j}$ are constants, subject to

$$A\mathbf{x} = \mathbf{b}$$

$$x_j \geqslant 0 \qquad j = 1 \ldots n$$

where

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)$$

$$\mathbf{b} = (b_1, b_2, \ldots, b_m)$$

and $A = (a_{ij})$ is an $m \times n$ matrix. Suppose also that we have m linearly independent columns of A which yield the following feasible linear combinations:

$$a_{.\ell_1}\bar{a}_{10} + a_{.\ell_2}\bar{a}_{20} + \cdots + a_{.\ell_m}\bar{a}_{m0} = \mathbf{b}$$

where

$$\bar{a}_{j0} \geqslant 0$$

when

$$\ell_j \,\varepsilon\, J^c = \{\ell_1, \ell_2, \ldots, \ell_m\} \subset \{1, \ldots, n\}$$

The $x_j = x_{\ell_j}$ are called the current basic variables. If we then set the values of the remaining variables equal to zero:

$$x_{k_i} = 0 \qquad \text{for } k_i \,\varepsilon\, J$$

we obtain a current basic feasible solution. The x_{k_i} are called the current non-basic variables. The value of the objective function for this current basic feasible solution is:

$$\bar{a}_{00} = \sum_{j=1}^{m} a_{0\ell_j}\bar{a}_{j0}$$

The columns corresponding to current non-basic variables can then be expressed as linear combinations of the columns corresponding to the current basic variables. This can conveniently be recorded in tableau form, i.e.,

$$\bar{a}_{.k_i} = \sum_{j=1}^{m} \bar{a}_{ji}\, a_{.\ell_j} \qquad (\text{synthetic } x_{k_i})$$

$$\mathbf{b} = \sum_{j=1}^{m} \bar{a}_{j0}\, a_{.\ell_j}$$

can be represented as

	b	x_{k_1}	x_{k_2}	....	x_{k_i}	....	$x_{k_{n-m}}$
x_0	$\bar{a}_{00}$	$\bar{a}_{01}$	$\bar{a}_{02}$	....	$\bar{a}_{0i}$	....	$\bar{a}_{0,n-m}$
X_1	$\bar{a}_{10}$	$\bar{a}_{11}$	$\bar{a}_{12}$	....	$\bar{a}_{1i}$	....	$\bar{a}_{1,n-m}$
.	.	.	.		.		.
.	.	.	.		.		.
.	.	.	.		.		.
.	.	.	.		.		.
.	.	.	.		.		.
X_m	$\bar{a}_{m0}$	$\bar{a}_{m1}$	$\bar{a}_{m2}$	....	$\bar{a}_{mi}$	....	$\bar{a}_{m,n-m}$

Appended to the tableau is a row of reduced costs denoted by $\bar{a}_0$, $\bar{a}_{0i}$ giving the decrease in objective function value when one unit of synthetic x_{k_i} in the current basic feasible solution is replaced by one unit of real x_{k_i}. If an arbitrary amount of synthetic x_{k_i} is replaced in this manner, the resulting solution may be infeasible or feasible but not a basic feasible solution. However, the simplex algorithm introduces as much as possible some currently non-basic variable x_{k_i}, which has a negative reduced cost, while still retaining feasibility. This results in a new basic feasible solution as follows. If the value of x_{k_i} is increased from 0 to v_{k_i} then to maintain feasibility:

$$a_{.\ell_1}\bar{a}_{10} + a_{.\ell_2}\bar{a}_{20} + \cdots + a_{.\ell_m}\bar{a}_{m0} + a_{.k_i}v_{k_i} = \mathbf{b}$$

the value of the basic variable X_p must change to:

$$\bar{a}'_{p0} = \bar{a}_{p0} - \bar{a}_{pi}\, v_{k_i}$$

If it is assumed that the problem has a finite solution, then at least one of the $\bar{a}_{pi}$ must be greater than zero. Since the value of all basic variables must be greater than or equal to zero, v_{k_i} can be increased until $\bar{a}'_{r0} = 0$ for some basic variable X_r. Then x_{ℓ_r} becomes a non-basic variable and $x_{k_i} = X_r$ enters the basis in the amount $\bar{a}_{r0} = v_{k_i}$. The value of the objective function is then increased from $\bar{a}_{00}$ to $\bar{a}_{00} - \bar{a}_{0i}\, v_{k_i}$ (recall that $\bar{a}_{0i} < 0$).

Since b is represented by a unique linear combination of the columns of A corresponding to the current basis, and since (in the non-degenerate case) the value of the objective function increases with each change to a new basic feasible solution, there must be a unique objective function value associated with each basic feasible solution. Hence, the algorithm can never return to the same basis twice and as the procedure is repeated, for problems assumed to have finite optimum, each basic feasible solution is examined until in a finite number of steps one is found for which all the reduced costs are positive. For this solution no currently non-basic variable can be introduced into the basis without decreasing the objective function value; hence, this must be the optimal solution.

B. Penalties

Suppose we have solved the following mixed integer linear program:

Maximize

$$x_0 = \sum_{j=1}^{n} a_{0j}x_j$$

subject to $Ax = b$, where A is an $(m \times n)$ matrix

$$x_j \geq 0$$

$$x_j = \text{integer} \qquad \text{for } j \in I \subset \{1 \ldots n\}$$

but as a linear program neglecting the integer constraints. Regardless of the LP method used in the solution the final simplex tableau can still be obtained. Let

$$\{x_{k_i}\} \qquad \text{for } k_i \in J \subset \{1 \ldots n\}$$

be the set of non-basic variables and

$$\{X_j = x_{\ell_j}\} \qquad \text{for } \ell_j \in J^c$$

be the set of basic variables. The tableau is then given as follows:

	b	x_{k_1}	x_{k_2}	....	x_{k_i}	....	$x_{k_{n-m}}$
x_0	$\bar{a}_{00}$	$\bar{a}_{01}$	$\bar{a}_{02}$	....	$\bar{a}_{0i}$	....	$\bar{a}_{0,n-m}$
X_1	$\bar{a}_{10}$	$\bar{a}_{11}$	$\bar{a}_{12}$	....	$\bar{a}_{1i}$	....	$\bar{a}_{1,n-m}$
.	.	.	.		.		.
.	.	.	.		.		.
.	.	.	.		.		.
.	.	.	.		.		.
.	.	.	.		.		.
X_m	$\bar{a}_{m0}$	..$\bar{a}_{m1}$	$\bar{a}_{m2}$	....	$\bar{a}_{mi}$	....	$\bar{a}_{m,n-m}$

(1) The column of A corresponding to the non-basic variable x_{k_i} is expressed as a linear combination of the columns of A corresponding to the basic variables (synthetic x_{k_i})

$$a_{.k_i} = \sum_{j=1}^{m} \bar{a}_{ji}\, a_{.\ell_j}$$

(2) The first column of the tableau gives the values of the objective function and the basic variables in the final solution

$$\bar{a}_{00} = \sum_{j=1}^{m} a_{0\ell_j}\bar{a}_{j0}$$

(3) If the non-basic variables x_{k_i} are changed in value from zero to v_{k_i}, then in order to maintain feasi-

bility the value of the basic variables X_p must change to

$$\bar{a}'_{p0} = \bar{a}_{p0} - \sum_{i=1}^{n-m} \bar{a}_{pi}\, v_{k_i} \qquad \text{for } p = 1 \ldots m$$

(4) The decrease in the objective function value when one unit of synthetic x_{k_i} in the optimal solution is replaced by one unit of x_{k_i} is given by $\bar{a}_{0i}$ (the reduced cost of x_{k_i}), and since the solution is optimal the reduced costs must all be positive.

Suppose that the basic variable X_p has the current optimal value $\bar{a}_{p0}$. If a new problem is created by appending to the current problem the constraint: $X_p \geqslant \bar{a}_{p0} + \alpha$ then the value of the objective function must decrease since the feasible region has been reduced. Since the number of constraints has been increased by one in the new problem, one of the currently non-basic variables x_{k_q} must enter the basis of the new problem.

If the value of α is small enough the rest of the basic variables will remain the same as in the current optimal solution. From the current tableau the minimum amount in which x_{k_q} may be introduced can be determined (as in 3 above)

$$\bar{a}'_{p0} = \bar{a}_{p0} - \bar{a}_{pq} v_{k_q} \geqslant \bar{a}_{p0} + \bar{\alpha}$$

$$v_{k_q} \geqslant -\frac{\alpha}{\bar{a}_{pq}}$$

since x_{k_q} must be introduced in a positive amount must have $a_{pq} < 0$

The decrease in objective function value from the current optimum when x_{k_q} is introduced in the amount v_{k_q} can then be determined from the reduced cost:

$$D \geqslant \bar{a}_{0q}\left[-\frac{\alpha}{\bar{a}_{pq}}\right]$$

Considering all of the non-basic variables which may enter the basis of the new problem in this manner, the current objective function value must be decreased by at least:

$$D_u = \alpha \min_{\substack{j \\ \bar{a}_{pj} < 0}} \left[-\frac{\bar{a}_{0j}}{\bar{a}_{pj}}\right]$$

and if this minimum is taken for $j = q$ then the amount of the non-basic variable x_{k_q} in the solution must increase from zero to

$$v_{k_q} = -\frac{\alpha}{\bar{a}_{pq}}$$

Similarly, if a new problem is created by appending to the current problem the further constraint: $x_p \leqslant a_{p0} - \beta$ the degradation of the objective function value must be at least

$$D_d = \beta \min_{\substack{j \\ \bar{a}_{pj} > 0}} \left[\frac{\bar{a}_{0j}}{\bar{a}_{pj}}\right]$$

and if this minimum is taken for $j = r$, then the amount of the non-basic variable x_{k_r} in the solution must increase from zero to

$$v_{k_r} = \frac{\beta}{\bar{a}_{pr}}$$

These results may now be applied to an integer-restricted basic variable x_p which has the current optimal value

$$\bar{a}_{p0} = n_{p0} + f_{p0}$$

where n_{p0} is an integer and $0 < f_{p0} < 1$.

Setting

$$\alpha = 1 - f_{p0}$$

(corresponding to adding the constraint $X_p \geqslant n_{p0} + 1$)

$$\beta = f_{p0}$$

(corresponding to adding the constraint $X_p \leqslant n_{p0}$)

the degradations of the objective functions must be at least

$$D_u = (1 - f_{p0}) \cdot \min_{\substack{j \\ \bar{a}_{pj} < 0}} \left[-\frac{\bar{a}_{0j}}{\bar{a}_{pj}}\right]$$

$$D_d = f_{p0} \cdot \min_{\substack{j \\ \bar{a}_{pj} > 0}} \left[\frac{\bar{a}_{0j}}{\bar{a}_{pj}}\right]$$

If the minima are taken for $j = q$ and $j = r$ respectively, then the amounts of the currently non-basic variables x_{k_q} and x_{k_r} must have increased in the optimal solutions of the new problems. However, if either x_{k_q} or x_{k_r} is an integer restricted variable, its amount in any integer solution of the new problem must have increased from its current zero value by an integer amount of at least one. The degradations of the current objective function value must therefore be at least equal to the reduced costs $\bar{a}_{0q}$ or $\bar{a}_{0r}$.

Thus, we can formulate two penalties which can be used to find upper bounds on the objective functions of the more tightly constrained problems created from the original problem.

$$P_u^p = \min_{\substack{j \\ \bar{a}_{pj}<0}} \begin{cases} (1-f_{p0})\left[-\dfrac{\bar{a}_{0j}}{\bar{a}_{pj}}\right] & j \notin I \quad (\text{i.e., } x_{k_j} \neq \text{integer}) \\ \max\left\{(1-f_{p0})\left[-\dfrac{\bar{a}_{0j}}{\bar{a}_{pj}}\right], \bar{a}_{0j}\right\} & j \in I \end{cases}$$

which gives as an upper bound on the objective function for the original problem with the appended constraint $X_p \leqslant n_{p0}$ the value

$$UB = \bar{a}_{00} - P_u^p$$

$$P_d^p = \min_{\substack{j \\ \bar{a}_{pj}>0}} \begin{cases} f_{p0}\left[\dfrac{\bar{a}_{0j}}{\bar{a}_{pj}}\right] & j \notin I \\ \max\left\{f_{p0}\left[\dfrac{\bar{a}_{0j}}{\bar{a}_{pj}}\right], \bar{a}_{0j}\right\} & j \in I \end{cases}$$

which gives as an upper bound on the objective function for the original problem with the appended constraint: $X_p \leqslant n_{p0}$ the value

$$UB - \bar{a}_{00} = P_d^p$$

A stronger upper bound on the value of the objective function of a subproblem which was obtained by more tightly constraining the current problem can be obtained by a Gomory cut. Gomory showed that if the integer-restricted variable X_p is unsatisfied in the LP solution of the current problem and has the value $\bar{a}_{p0} = n_{p0} + f_{p0}$, then any feasible integer solution of the current problem (and hence any feasible integer solution of a more tightly constrained subproblem) must satisfy the following additional inequality:

$$-f_{p0} - \sum_{j=1}^{n} f_{pj}^{*}(-x_j) \geqslant 0$$

where

$$\bar{a}_{pj} = n_{pj} + f_{pj}$$

and

$$f_{pj}^{*} = \begin{cases} \bar{a}_{pj} & \bar{a}_{pj} \geqslant 0 \text{ and } j \notin I \\ \dfrac{f_{p0}(-\bar{a}_{pj})}{(1-f_{p0})} & \bar{a}_{pj} < 0 \text{ and } j \notin I \\ f_{pj} & f_{pj} \leqslant f_{p0} \text{ and } j \in I \\ \dfrac{f_{p0}(1-f_{pj})}{(1-f_{p0})} & f_{pj} > f_{p0} \text{ and } j \in I \end{cases}$$

If this inequality is appended as a constraint to the current problem, then the degradation of the objective function for this subproblem from the dual-simplex method must be at least

$$D_G = \min_{j \in \{1 \ldots n\}} \bar{a}_{0j}\left[\frac{f_{p0}}{f_{pj}^{*}}\right]$$

Using this value it is now possible to give a penalty for satisfying the integer requirement of any currently unsatisfied basic integer restricted variable:

$$P_G^p = \min_{j \in \{1 \ldots n\}} \begin{cases} f_{p0}\dfrac{\bar{a}_{0j}}{\bar{a}_{pj}} & \bar{a}_{pj} \geqslant 0 \text{ and } j \notin I \\ (1-f_{p0})\left[-\dfrac{\bar{a}_{0j}}{\bar{a}_{pj}}\right] & \bar{a}_{pj} < 0 \text{ and } j \notin I \\ \bar{a}_{0j}\dfrac{f_{p0}}{f_{pj}} & f_{pj} \leqslant f_{p0} \text{ and } j \in I \\ \bar{a}_{0j}\dfrac{(1-f_{p0})}{(1-f_{pj})} & f_{pj} > f_{p0} \text{ and } j \in I \end{cases}$$

An upper bound on the value of the objective function of any integer solution attainable from the current problem is then given by

$$UB = \bar{a}_{00} - P_G^p$$

A choice may now be made, based on these penalties, as to which unsatisfied integer restricted variable is to be used for branching. The most commonly followed procedures are as follows:

1) Calculate the penalties P_n^p and P_d^p for all currently unsatisfied integer restricted variables.
   - –Choose the variable associated with the smallest penalty as the branching variable.
   - –Create two new subproblems by appending to the current problem the constraints: $X_p \geqslant n_{p0} + 1$ and $X_p \leqslant n_{p0}$

- Choose the new subproblem associated with the smaller penalty for immediate solution.

- Place the other new subproblem in LIST with its upper bound (determined from P_u^p or P_d^p and P_G^p).

This procedure is based on the assumption that in most cases the smaller penalty will reflect the smaller true degradation, and hence that an integer solution will be reached along this branch with a high objective function value. If this assumption is not justified, a refinement of the above procedure known as node swapping can be used:

1a) Compare the true degradation of the subproblem which was solved immediately with the penalty of the postponed subproblem.

- If the true degradation exceeds the penalty, solve the postponed subproblem to find its true degradation, then choose the subproblem with the lower true degradation as the problem from which to continue branching.

- If the true degradation of the immediately solved subproblem is less than the penalty of the postponed subproblem, then the subproblem solved must have the lower true degradation and is the problem from which branching should continue.

Or better still

2) Calculate the penalties P_u^p and P_d^p for all currently unsatisfied integer-restricted variables.

- Choose the variable associated with the largest penalty as the branching variable.

- Create two new subproblems by appending to the current problem the constraints: $X_p \geqslant n_{p0} + 1$ and $X_p \leqslant n_{p0}$

- Place the new subproblem associated with the larger penalty in LIST with its upper bound (determined from P_u^p or P_d^p and P_G^p).

- Choose the other new subproblem for immediate solution.

This procedure has the advantage of postponing the problems which are known to have the smallest objective function values until later in the search when presumably there will be an integer solution available with a larger objective function value in which case the postponed subproblem need not be solved at all since it could not possibly be optimal.

C. Shortcomings of the Penalty Approach

In cases where the number of constraints is very large, however, the penalty method of directing the search for an optimal integer solution breaks down and the search becomes essentially random. The reason for this is that a calculated penalty will not represent in any manner the true degradation of the associated subproblem. In fact, the larger penalty may be in the direction of the much smaller true degradation. This is illustrated in Fig. A-1 where the feasible set of the current problem has been projected on the (X_p, x_0) plane.

To direct the search in these cases, branching can be based on priorities. The branching variable is chosen as the unsatisfied integer-restricted variable in the solution of the current problem which is highest on a priority list supplied exogenously by the user. The postponement or solution of the newly created subproblems is then based on penalties (perhaps with node swapping). The priority list may be determined from the user's knowledge of which variables will have the greatest effect on the overall system or, failing this, priorities may be assigned in order of the cost coefficient values in the original objective function.

See the Level 2 flowcharts in Fig. A-1 that describe the above methods (note that the flowchart for BRANCH1A utilizes the flowchart for BRANCH1 as a Level 3 flowchart)

II. BACKTRAK

In most cases the procedure of further constraining unsatisfied integer-restricted variables will eventually lead to a point where the subproblem chosen for branching cannot usefully be further constrained. This can happen if it becomes infeasible or if its value falls below the value of the best integer solution currently available or if it yields a new best integer solution. In these cases it is necessary to have a procedure for choosing a problem from the stored list from which to continue the search. The earliest such procedure used was LIFO (last in–first out) in which the next problem chosen was the last problem placed in the list which has an upper bound greater than the current best integer solution value and was dictated by the serial access nature of the storage devices available when it was first implemented.

In general, such a choice of procedure may be far from optimal. An improved procedure is to choose as the next problem the problem in the stored list which has the largest upper bound on its objective function value. This procedure takes advantage of the newer random access storage devices but still not in the most efficient manner. The disadvantage of this procedure is that it takes into account only the objective function value and excludes other, perhaps equally important, properties of the stored problems: primarily, the amount of work necessary to bring the chosen problem to an integer solution. A current method which takes both of these factors into account is the best projection criterion.

A. Best Projection Criterion

Let the optimal value of the objective function for the original mixed-integer problem with the integer requirements relaxed be x_0^0 and the value of the objective function of the latest integer solution found be x_0^I. If the first integer solution has not been reached, some estimate, possibly inaccurate, of the value of x_0^I may be given. Define the sum of integer infeasibilities

$$s = \sum_{\substack{p=1 \\ I_p \in I}}^{m} \min\{f_{po}, 1 - f_{po}\}$$

as a measure of how much the integer restricted variables in the problem differ from integer values in the solution. If the objective function value x_0^k of each outstanding problem presently stored in the list is plotted against its sum of integer infeasibilities s^k and then projected parallel to the line between (s^0, x_0^0) and $(0, x_0^I)$ onto the line $s = 0$ we get as the projected value

$$p_k = x_0^k - \frac{x_0^0 - x_0^I}{s^0} s^k$$

See Fig. A-3.

Here

$$\lambda = \frac{x_0^0 - x_0^I}{s^0}$$

gives an estimate of the marginal degradation of the objective function value for a unit decrease in the sum of integer infeasibilities and hence p_k is an estimate of the objective function value which can be obtained in an integer solution ($s = 0$) attainable from the current outstanding problem k and is of course more accurate when s^k is small. The potential objective function value of a problem can be estimated from its upper bound while an estimate of its sum of integer infeasibilities can be obtained from its value in the parent problem which branched to yield the stored problem. The next problem chosen for solution is then the one with the largest projected integer solution value p^k. If the value of λ is overestimated, more weight is placed on objective function value in deciding which outstanding problem is to be chosen next for solution. If the value of λ is underestimated, more weight is placed on the proximity to an integer solution in this decision.

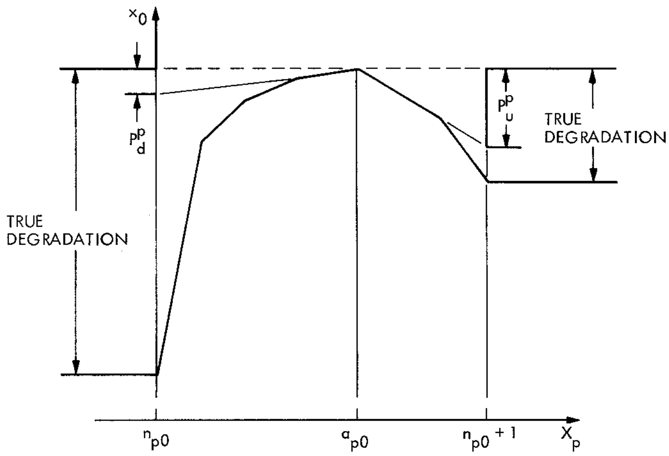


Fig. A-1. A larger penalty in the direction of smaller degradation

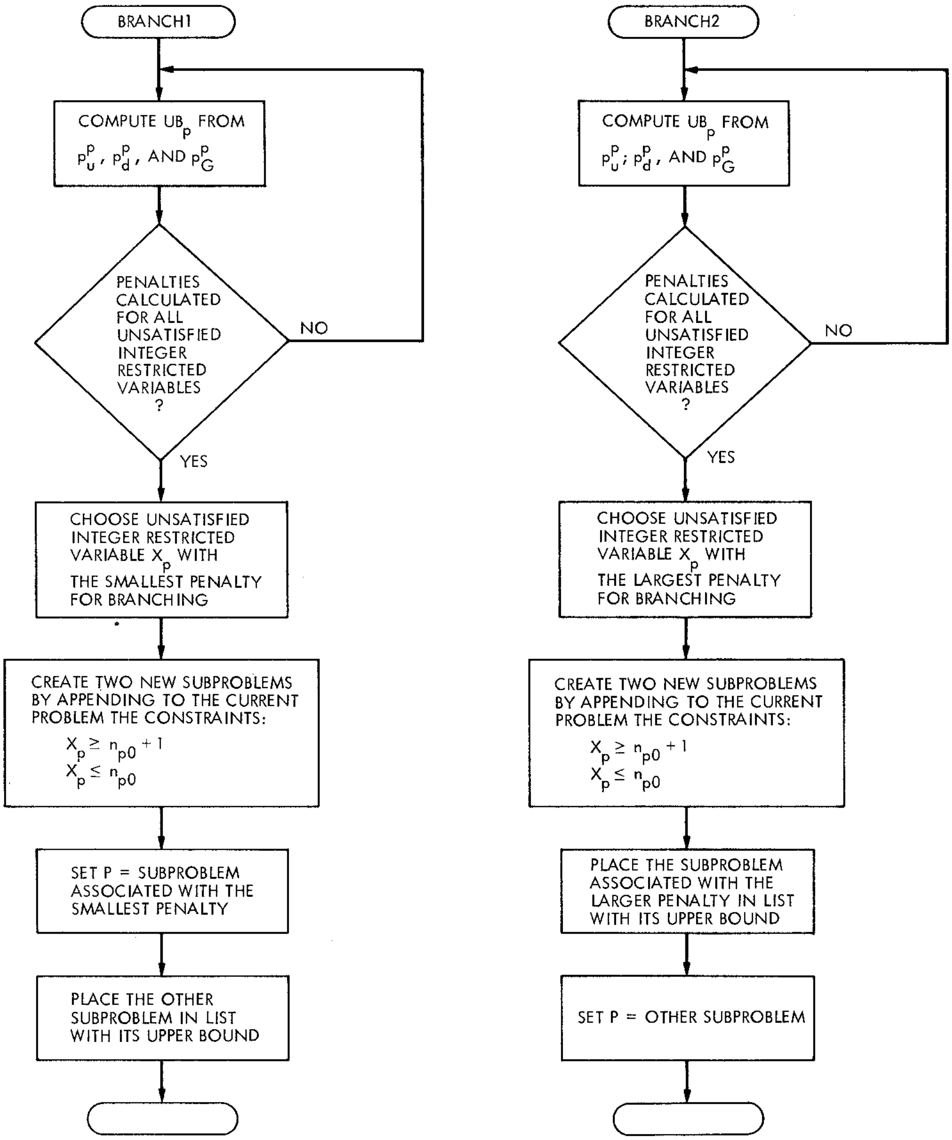


Fig. A-2. Flowcharts for BRANCH1, BRANCH2, and BRANCH1A

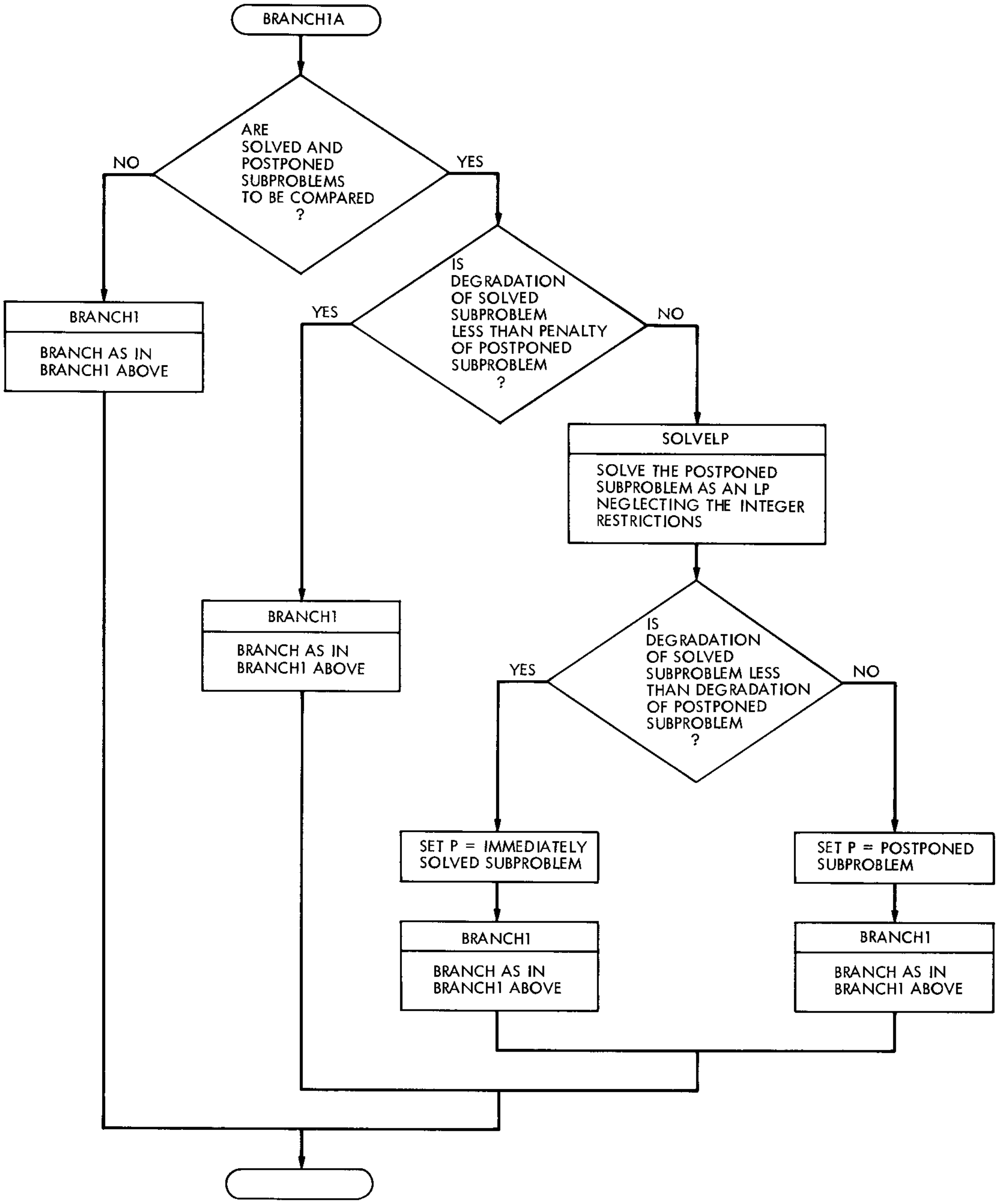

BRANCH1A
ARE SOLVED AND POSTPONED SUBPROBLEMS TO BE COMPARED ?
NO
YES
BRANCH1
BRANCH AS IN BRANCH1 ABOVE
IS DEGRADATION OF SOLVED SUBPROBLEM LESS THAN PENALTY OF POSTPONED SUBPROBLEM ?
YES
NO
SOLVELP
SOLVE THE POSTPONED SUBPROBLEM AS AN LP NEGLECTING THE INTEGER RESTRICTIONS
BRANCH1
BRANCH AS IN BRANCH1 ABOVE
IS DEGRADATION OF SOLVED SUBPROBLEM LESS THAN DEGRADATION OF POSTPONED SUBPROBLEM ?
YES
NO
SET P = IMMEDIATELY SOLVED SUBPROBLEM
SET P = POSTPONED SUBPROBLEM
BRANCH1
BRANCH AS IN BRANCH1 ABOVE
BRANCH1
BRANCH AS IN BRANCH1 ABOVE

Fig. A-2 (contd)

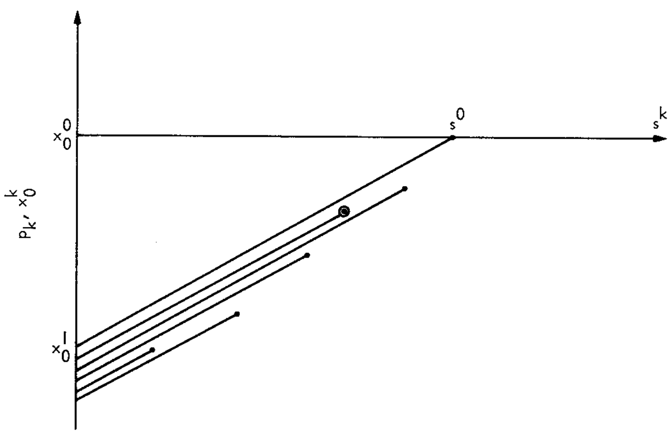


Fig. A-3. Objective function versus sum of infeasibilities

NASA — JPL — Coml., L.A., Calif.